Propositional and First-Order Logic

Chapter 7.4─7.8, 8.1─8.3, 8.5

Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer

Logic roadmap overview

- Propositional logic (review)
- Problems with propositional logic
- First-order logic (review)
  - Properties, relations, functions, quantifiers, …
  - Terms, sentences, wffs, axioms, theories, proofs, …
- Extensions to first-order logic
- Logical agents
  - Reflex agents
  - Representing change: situation calculus, frame problem
  - Preferences on actions
  - Goal-based agents

Disclaimer

"Logic, like whiskey, loses its beneficial effect when taken in too large quantities."

- *Lord Dunsany*

Propositional Logic: Review

Big Ideas

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic** is the simple foundation and fine for some AI problems
- **First order logic** (FOL) is much more expressive as a KR language and more commonly used in AI
- There are many variations: horn logic, higher order logic, three-valued logic, probabilistic logics, etc.

Propositional logic

- **Logical constants**: true, false
- **Propositional symbols**: P, Q,... (aka **atomic sentences**)
- Wrapping **parentheses**: (…)
- Sentences are combined by **connectives**:

$\wedge$	and	[conjunction]
$\vee$	or	[disjunction]
$\Rightarrow$	implies	[implication / conditional]
$\Leftrightarrow$	is equivalent	[biconditional]
$\neg$	not	[negation]

- **Literal**: atomic sentence or negated atomic sentence: P, $\neg$ P

Examples of PL sentences

- $(P \wedge Q) \rightarrow R$

  "If it is hot and humid, then it is raining"
- $Q \rightarrow P$

  "If it is humid, then it is hot"
- Q

  "It is humid."
- We're free to choose better symbols, btw:

  Ho = "It is hot"

  Hu = "It is humid"

  R = "It is raining"

Propositional logic (PL)

- Simple language for showing key ideas and definitions
- User defines set of propositional symbols, e.g., P, Q
- User defines **semantics** of each propositional symbol:
  - P means "It is hot", Q means "It is humid", etc.
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg$S is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then (S $\vee$ T), (S $\wedge$ T), (S $\rightarrow$ T), and (S $\leftrightarrow$ T) are sentences
  - A sentence results from a finite number of applications of the rules

Some terms

- The meaning or **semantics** of a sentence determines its **interpretation**
- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its **truth value** (True or False)
- A **model** for a KB is a *possible world* – an assignment of truth values to propositional symbols that makes each sentence in the KB True

Model for a KB

- Let the KB be [$P \wedge Q \rightarrow R$, $Q \rightarrow P$]
- What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables

  PQR

  - **FFF: OK**
  - **FFT: OK**
  - FTF: NO
  - FTT: NO
  - **TFF: OK**
  - **TFT: OK**
  - TTF: NO
  - **TTT: OK**

P: it's hot
Q: it's humid
R: it's raining

- If KB is [$P \wedge Q \rightarrow R$, $Q \rightarrow P$, Q], the **only** model is TTT

More terms

- A **valid sentence** or **tautology** is a sentence that's True under all interpretations, no matter what the world is actually like or what the semantics is. Example: "It's raining or it's not raining"
- An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in "It's raining and it's not raining."
- **P entails Q**, written P |= Q, means that whenever P is True, so is Q
  - In all models in which P is true, Q is also true

Truth tables

- Truth tables are used to define logical connectives
- And to determine when a complex sentence is true given the values of the symbols in it

Truth tables for the five logical connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Example of a truth table used for a complex sentence

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

On the implies connective: $P \rightarrow Q$

- Note that $\rightarrow$ is a logical connective
- So $P \rightarrow Q$ is a logical sentence and has a truth value, i.e., is either true or false
- If we add this sentence to the KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove Q if P is also in the KB
- Given a KB where P=True and Q=True, we can also derive/infer/prove that $P \rightarrow Q$ is True

$P \rightarrow Q$

- When is $P \rightarrow Q$ true? Check all that apply
  - ❑ P=Q=true
  - ❑ P=Q=false
  - ❑ P=true, Q=false
  - ❑ P=false, Q=true

P → Q

- When is *P→Q* true? Check all that apply
  - ☑ P=Q=true
  - ☑ P=Q=false
  - ☐ P=true, Q=false
  - ☑ P=false, Q=true
- We can get this from the truth table for →
- Note: in FOL it's much harder to prove that a conditional true
  - Consider proving prime(x) → odd(x)

Inference rules

- **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
- An inference rule is **sound** if every sentence X it produces when operating on a KB logically follows from the KB
  - i.e., inference rule creates no contradictions
- An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB.
  - Note analogy to complete search algorithms

Sound rules of inference

- Here are some examples of sound rules of inference
- Each can be shown to be sound using a truth table

RULE	PREMISE	CONCLUSION
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$A \vee B, \neg B \vee C$	$A \vee C$

Soundness of modus ponens

A	B	$A \rightarrow B$	OK?
True	True	True	√
True	False	False	√
False	True	True	√
False	False	True	√

Resolution

- **Resolution** is a valid inference rule producing a new clause implied by two clauses containing *complementary literals*
  - A literal is an atomic symbol or its negation, i.e., P, ~P
- Amazingly, this is the only interference rule you need to build a sound and complete theorem prover
  - Based on proof by contradiction and usually called resolution refutation
- The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 1960s

Resolution

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.
- To use resolution, put KB into *conjunctive normal form* (CNF) where each is a disjunction of (one or more) literals (positive or negative atoms)
- Every KB can be put into CNF, it's just a matter of rewriting its sentences using standard tautological rules
  - $(A \rightarrow B) \leftrightarrow (\sim A \vee B)$
  - $(A \vee (B \wedge C)) \leftrightarrow (A \vee B) \wedge (A \vee C)$
  - $A \wedge B \rightarrow A$
  - $A \wedge B \rightarrow B$

Resolution Example

- KB: [$P \rightarrow Q$, $Q \rightarrow R \wedge S$]
- KB in CNF: [$\sim P \vee Q$, $\sim Q \vee R$, $\sim Q \vee S$]
- Resolve KB(1) and KB(2) producing:

  $\sim P \vee R$ *(i.e., $P \rightarrow R$)*
- Resolve KB(1) and KB(3) producing:

  $\sim P \vee S$ *(i.e., $P \rightarrow S$)*
- New KB: [$\sim P \vee Q$, $\sim Q \vee R$, $\sim Q \vee S$, $\sim P \vee R$, $\sim P \vee S$]

Tautologies

$(A \rightarrow B) \leftrightarrow (\sim A \vee B)$

$(A \vee (B \wedge C)) \leftrightarrow (A \vee B) \wedge (A \vee C)$

Soundness of the resolution inference rule

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

From the rightmost three columns of this truth table, we can see that

$$(\alpha \vee \beta) \wedge (\sim\beta \vee \gamma) \leftrightarrow (\alpha \vee \gamma)$$

is valid (i.e., always true regardless of the truth values assigned to α, β and γ

Proving things

- A **proof** is a sequence of sentences, where each is a premise (i.e., a given) or is derived from earlier sentences in the proof by an inference rule
- The last sentence is the **theorem** (also called goal or query) that we want to prove
- Example for the "weather problem"

1 Hu	premise	"It's humid"
2 $Hu \rightarrow Ho$	premise	"If it's humid, it'shot"
3 Ho	modus ponens(1,2)	"It's hot"
4 $(Ho \wedge Hu) \rightarrow R$	premise	"If it's hot & humid, it's raining"
5 $Ho \wedge Hu$	and introduction(1,3)	"It's hot and humid"
6 R	modus ponens(4,5)	"It's raining"

Horn* sentences

- A **Horn sentence** or **Horn clause** has the form:

  $P1 \wedge P2 \wedge P3 \ldots \wedge Pn \rightarrow Qm$ *where n>=0, m in{0,1}*

- Note: a conjunction of 0 or more symbols to left of $\rightarrow$ and 0-1 symbols to right
- Special cases:
  - n=0, m=1: P (assert P is true)
  - n>0, m=0: $P \wedge Q \rightarrow$ (constraint: both P and Q can't be true)
  - n=0, m=0: (well, there is nothing there!)
- Put in CNF: each sentence is a disjunction of literals with at most one non-negative literal

  $\neg P1 \vee \neg P2 \vee \neg P3 \ldots \vee \neg Pn \vee Q$

($P \rightarrow Q) = (\neg P \vee Q)$

* After Alfred Horn

Significance of Horn logic

- We can also have horn sentences in FOL
- Reasoning with horn clauses is much simpler
  - Satisfiability of a propositional KB (i.e., finding values for a symbols that will make it true) is NP complete
  - Restricting KB to horn sentences, satisfiability is in P
- For this reason, FOL Horn sentences are the basis for many rule-based languages, including Prolog and Datalog
- Horn logic gives up handling, in a general way, (1) negation and (2) disjunctions

Entailment and derivation

- **Entailment: KB |= Q**
  - Q is entailed by KB (set sentences) iff there is no logically possible world where Q is false while all the sentences in KB are true
  - Or, stated positively, Q is entailed by KB iff the conclusion is true in every logically possible world in which all the premises in KB are true
- **Derivation: KB |- Q**
  - We can derive Q from KB if there's a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

Two important properties for inference

Soundness: If KB |- Q then KB |= Q

- If Q is derived from KB using a given set of rules of inference, then Q is entailed by KB
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid

Completeness: If KB |= Q then KB |- Q

- If Q is entailed by KB, then Q can be derived from KB using the rules of inference
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises

Problems with Propositional Logic

Propositional logic: pro and con

- **Advantages**
  - Simple KR language sufficient for some problems
  - Lays the foundation for higher logics (e.g., FOL)
  - Reasoning is decidable, though NP complete, and efficient techniques exist for many problems
- **Disadvantages**
  - Not expressive enough for most problems
  - Even when it is, it can very “un-concise”

PL is a weak KR language

- Hard to identify "individuals" (e.g., Mary, 3)
- Can't directly talk about properties of individuals or relations between individuals (e.g., "Bill is tall")
- Generalizations, patterns, regularities can't easily be represented (e.g., "all triangles have 3 sides")
- First-Order Logic (FOL) is expressive enough to represent this kind of information using relations, variables and quantifiers, e.g.,
  - *Every elephant is gray:* ∀ x (elephant(x) → gray(x))
  - *There is a white alligator:* ∃ x (alligator(X) ^ white(X))

PL Example

- Consider the problem of representing the following information:
  - Every person is mortal.
  - Confucius is a person.
  - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?

PL Example

- In PL we have to create propositional symbols to stand for all or part of each sentence, e.g.:

  P = "person"; Q = "mortal"; R = "Confucius"
- The above 3 sentences are represented as:

  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$
- The 3rd sentence is entailed by the first two, but we need an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes *person* and *mortal*
- Representing other individuals requires introducing separate symbols for each, with some way to represent the fact that all individuals who are "people" are also "mortal"

Hunt the Wumpus domain

- Some atomic propositions:

  S12 = There is a stench in cell (1,2)

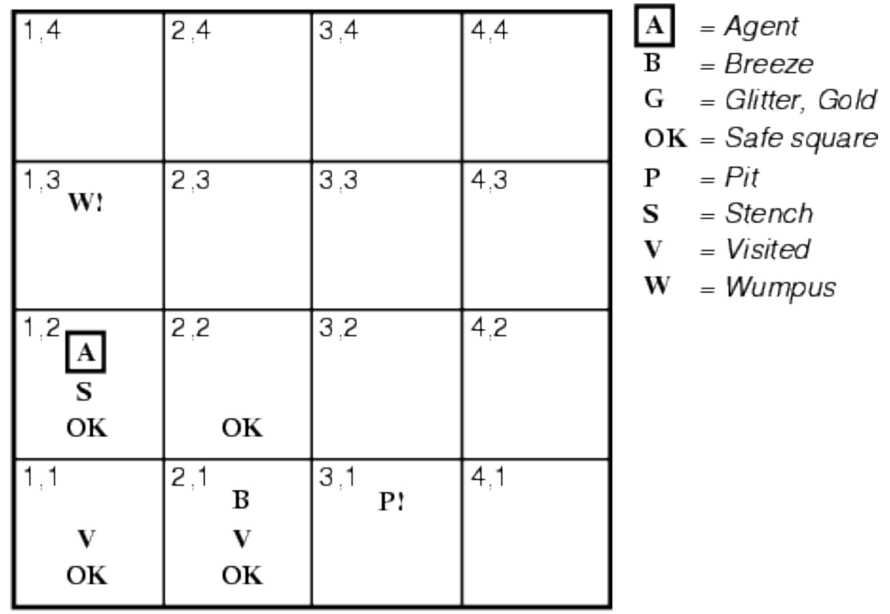
  B34 = There is a breeze in cell (3,4)

  W22 = Wumpus is in cell (2,2)

  V11 = We've visited cell (1,1)

  OK11 = Cell (1,1) is safe

  …



- Some rules:

  ¬S22 → ¬W12 ∧ ¬W23 ∧ ¬W32 ∧ ¬W21

  S22 → W12 ∨ W23 ∨ W32 ∨ W21

  B22 → P12 ∨ P23 ∨ P32 ∨ P21

  W22 → S12 ∧ S23 ∧ S23 ∧ W21

  W22 → ¬W11 ∧ ¬W21 ∧ … ¬W44

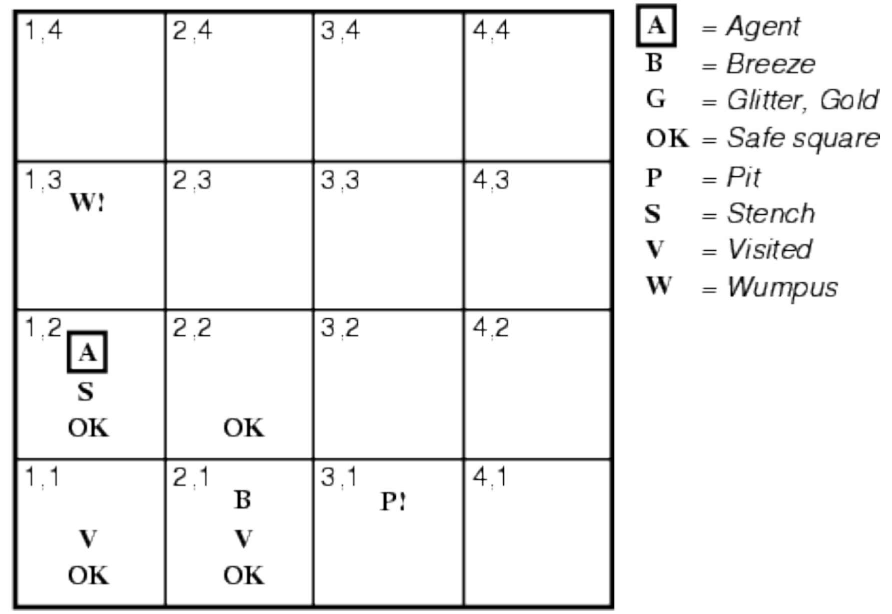
  A22 → V22

  A22 →¬W11 ∧ ¬W21 ∧ … ¬W44

  V22 → OK22

Hunt the Wumpus domain

- Eight variables for each cell: e.g., A11, B11, G11, OK11, P11, S11, V11, W11
- The lack of variables requires us to give similar rules for each cell!
- Ten rules (I think) for each



A11 → …

V11 → …

P11 → …

¬P11 → …

W11 → …

¬W11 → …

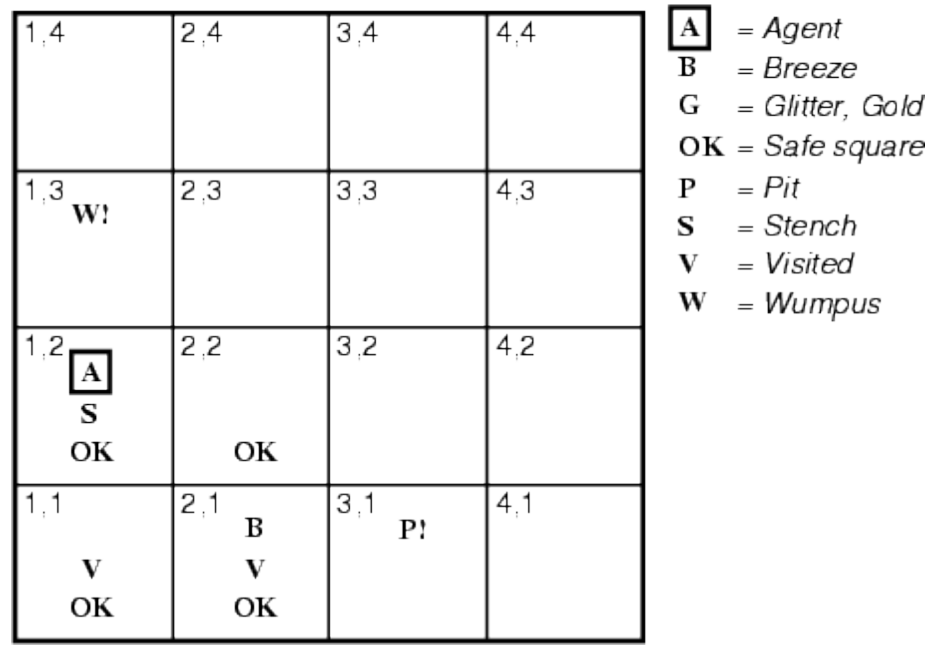
S11 → …

¬S11 → …

B11 → …

¬B11 → …

After third move



- We can prove that the Wumpus is in (1,3) using these four rules
- See R&N section 7.5

$$(R1)\ \neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$$

$$(R2)\ \neg S21 \rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$$

$$(R3)\ \neg S12 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$$

$$(R4)\ \ S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$$

Proving W13

(R1) ¬S11 → ¬W11 ∧ ¬ W12 ∧ ¬ W21

(R2) ¬ S21 → ¬W11 ∧ ¬ W21 ∧ ¬ W22 ∧ ¬ W31

(R3) ¬ S12 → ¬W11 ∧ ¬ W12 ∧ ¬ W22 ∧ ¬ W13

(R4) S12 → W13 ∨ W12 ∨ W22 ∨ W11

Apply MP with ¬S11 and R1:

¬ W11 ∧ ¬ W12 ∧ ¬ W21

Apply And-Elimination to this, yielding 3 sentences:

¬ W11, ¬ W12, ¬ W21

Apply MP to ~S21 and R2, then apply And-elimination:

¬ W22, ¬ W21, ¬ W31

Apply MP to S12 and R4 to obtain:

W13 ∨ W12 ∨ W22 ∨ W11

Apply Unit Resolution on (W13 ∨ W12 ∨ W22 ∨ W11) and ¬W11:

W13 ∨ W12 ∨ W22

Apply Unit Resolution with (W13 ∨ W12 ∨ W22) and ¬W22:

W13 ∨ W12

Apply Unit Resolution with (W13 ∨ W12) and ¬W12:

W13

QED

Propositional Wumpus hunter problems

- Lack of variables prevents stating more general rules
  - $\forall$ x, y V(x,y) $\rightarrow$ OK(x,y)
  - $\forall$ x, y S(x,y) $\rightarrow$ W(x-1,y) $\vee$ W(x+1,y) …
- Change of the KB over time is difficult to represent
  - In classical logic, a fact is true or false for all time
  - A standard technique is to index dynamic facts with the time when they're true
    - A(1, 1, t0)
  - Thus we have a separate KB for every time point

Propositional logic summary

- Inference: process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives all true conclusions from a set of premises
- **Valid sentence:** true in all worlds under all interpretations
- If an implication sentence can be shown to be valid, then, given its premise, its consequent can be derived
- Different logics make different **commitments** about what the world is made of and the kind of beliefs we can have
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffices to illustrate the process of inference
  - Propositional logic can become impractical, even for very small worlds